



# **PILVIPEITTOISUUDEN ARVIOIMINEN METAR- DATASTA**

Jarno Rannanpää

Ohjaaja: Timo Rahkonen

**ELEKTRONIIKAN JA TIETOLIIKENNETEKNIIKAN  
TUTKINTO-OHJELMA**

**2022**

**Rannanpää J. J. (2022) Pilvipeittoisuuden arvioiminen METAR-datasta.** Oulun yliopisto, Elektroniikan ja tietoliikennetekniikan tutkinto-ohjelma. Kandidaatintyö, 31 s.

## **TIIVISTELMÄ**

Työn tarkoituksena oli arvioida paikallista pilvipeitteisyyttä Oulun lentoaseman tuottamasta METAR-datasta käyttäen AWK-kieltä sekä UNIX-pohjaisia komentorivityökaluja. Kahdesti tunnissa lähetettäviä METAR-tiedotteita analysoitiin vuosikymmenen 2010-2020 ajalta. Käsiteltävien rivien määrä liikkui sadoissa tuhansissa. Suuren datamäärän käsittely asetti omat vaatimuksensa tehokkaiden työkalujen käytölle.

Työn alussa käydään läpi METAR-viestien rakenne. Lisäksi selvitetään pilvipeitteisyyden laskeminen sekä itse laskemisen tekninen haaste. Viestit on alun perin suunniteltu ihmissilmän luettavaksi, eivätkä ne sen vuoksi noudata tarkkaa rakennetta tai mittaa, vaikka informaation järjestys pääsääntöisesti onkin vakioitu. Mm. pilvipeitteisyys on koodattu viesteihin kirjainlyhentein ja ilmoitettu eri korkeuksien osapeitteinä. Nämä osapeitteisyydet tuli paikantaa erimittaisista viesteistä sekä laskea edelleen kokonaispilvipeitteisyydeksi. Työn oleellisena osana oli selvittää AWK-kielen ja tyyppisten UNIX-pohjaisten komentorivityökalujen käyttöä tällaisen haasteen ratkaisemiseksi.

Konstruktio-osassa esitellään tarvittava koodaus pilvipeitteisyyden arvioimiseksi. Kaikki tarvittava koodaus on esitetty työn liitteissä. Työn viimeisessä käytännön osassa jouduttiin perehtymään myös epätäydellisen datan korjaamiseen; alkuperäisiä METAR-tiedotteita puuttui (jääneet mahdollisesti lähettämättä esimerkiksi huoltokatkoksen takia), jonka lisäksi esiintyi saman viestin monikertoja. Lopuksi kehitetyn koodauksen jatkokäyttöä arvioitiin. Koodin ja sen toiminnan lisäksi esitettiin graafinen esimerkkiesitys pilvisyydestä.

Työhön sovelletut työkalut osoittautuivat erittäin tehokkaiksi suuren datamäärän käsittelyyn sekä siinä toistuvien rakenteiden paikantamiseen, olkoonkin, että itse data oli rakenteeltaan hyvin epäsäännöllistä.

**Avainsanat:** METAR, pilvipeite, avoin data, UNIX, AWK.

**Rannanpää J. J. (2022) Total cloudiness estimation based on METAR data.**  
Degree Programme in Electronics and Communications Engineering, Bachelor's  
Thesis, 31 p.

## **ABSTRACT**

**The work aimed to evaluate local total cloudiness by using open data – in this case METAR messages from Oulu airport which are sent twice per hour. Data set of one whole decade 2010-2020 was analyzed. The tools to be used were specified being AWK language and typical UNIX shell tools. The number of message lines to be handled within the data was hundreds of thousands. Thus, specific tools for large amount of data were needed.**

**In the beginning of the work report, the structure of METAR messages is explained. Then, the logic how the total cloudiness is calculated is explored and the practical technical challenge further explained. In brief, sky coverage is coded with letter-based acronyms for several sky levels in messages that do not follow strict length nor structure; the order of the information typically still remains the same - respectively. These sub-levels were needed to be located throughout the data set and then further calculated as a total cloudiness of the sky. Core part of the work was to evaluate and execute AWK language and UNIX-based shell tools to solve the problem-field.**

**In the construction part of the work the needed coding to calculate total cloudiness out of the data set is presented. All needed coding is given in appendixes. In this last part of the work, non-perfect storage (or lack of existence) of data was observed (possibly due to messages never sent due to service breaks, etc.). Multiple entries of same data also existed. External time data set was used to help to fix the found imperfection. The function of fundamental parts of the code are explained and possible further usage of cloudiness information evaluated for possible further studies to come. Finally, graphical example is represented for better understanding how the cloudiness data could be applied.**

**The tools used to solve the problem were proven to be effective for the purpose, especially, for the big amount of data and for locating repeated structures from the data set. This, regardless of the data being irregular and with changing lengths.**

**Key words: METAR, sky coverage, cloudiness, open data, UNIX, AWK.**

## SISÄLLYS

|  |    |
|--|----|
| .....  | 1  |
| TIIVISTELMÄ.....   | 2  |
| ABSTRACT .....   | 3  |
| SISÄLLYS.....  | 4  |
| ALKULAUSE .....  | 5  |
| 1. JOHDANTO.....   | 6  |
| 2. METAR-sanoma .....                                    | 7  |
| 2.1. Mikä on METAR-sanoma? .....                         | 7  |
| 2.2. Muista sanomatyypeistä .....                        | 7  |
| 2.3. METAR-tiedotteen rakenne .....                      | 8  |
| 2.4. Taivaan peittoisuus.....                            | 9  |
| 3. Aineisto .....  | 12 |
| 3.1. METAR-tiedotteet.....                               | 12 |
| 3.2. Aurinkodata sekä korjaava aikasarja.....            | 13 |
| 4. Datan käsittely ja pohdinta .....                     | 15 |
| 4.1. Käsittelyn vaiheet.....                             | 15 |
| 4.2. METAR-tiedotteiden valmistelu ja esikäsittely ..... | 15 |
| 4.3. Aurinkodata ja METAR-datan puutteet .....           | 16 |
| 4.4. METAR-datan ja aurinkodatan yhdistäminen.....       | 17 |
| 4.5. Pohdinta.....                                       | 22 |
| 4.6. Jatko- ja parannuskehitysmahdollisuudet.....        | 23 |
| 5. YHTEENVETO .....                                      | 24 |
| 6. LÄHTEET .....   | 25 |
| 7. LIITTEET.....   | 26 |

## ALKULAUSE

Martille ja Severille. Jaksoitte aina käydessäni kysyä, miten opinnot sujuivat. Ette kaikeksi harmikseni ehtineet nähdä minun valmistuvan ja hoitavan opintojani loppuun.

Tämän työn tekemisessä otettiin haasteeksi täysin uuden taidon opettelu. UNIX-osaaminen ei alkujaan kuulunut henkilökohtaisesti vahvimpaan repertuaariini. Työ pohjasi myös käytännön tarpeeseen. Suurien mittausdatamäärien käsittely ja etenkin puutteellisen datan korjaus ovat hyödyllisiä työkaluja kaikille tämän alan käytännön insinöörintyötä tekeville.

Kiitokset menevät työn ohjaajalle Timo Rahkoselle. Johdatit minut mielenkiintoisten työkalujen pariin ja kannustit työn loppuun suorittamisessa. Kiitos myös perheelleni – etenkin lapsille – kärsivällisyydestä. Ehdittiin ne iltasadutkin lukemaan, vaikka kandidaatintyö iltapainotteisesti hiljalleen valmistuikin.

Oulussa, huhtikuussa 2022

Jarno Rannanpää

## 1. JOHDANTO

Internetistä löytyy valtavasti niin kutsuttua avointa dataa, jolla tarkoitetaan julkishallintojen, yksityissektorin tai ylipäätään erilaisten organisaatioiden tuottamaa dataa, joka on tuotu julkisesti kaikkien saataville – tyypillisesti hallinnoimilleen Internet-sivustoille. Hyviä esimerkkejä avoimesta datasta ovat esimerkiksi Ilmatieteenlaitoksen tuottamat säähavaintosarjat tai vaikkapa lentoasemien tuottamat METAR-sanomat.

Yksittäinen METAR-sanoma on kiinnostava paikalliselle lennonsuunnittelulle juuri nyt ja maksimissaan puoleksi tunniksi eteenpäin (frekvenssi esimerkiksi Oulun lentoasemalla). Tämä on myös niiden pääasiallinen käyttötarkoitus ja siksi etsiessä esimerkiksi Oulun lentoaseman METAR-sanomia, löytää ainoastaan tällä hetkellä voimassa olevan tiedotteen. Tämän kandidaatin työn kannalta oltiin kiinnostuneita, voitaisiinko suuresta määrästä METAR-sanomia laskea ulos kokonaispilvisyystieto käyttäen apuna sanomien mukaisia, eri korkeuksille määritettyjä taivaan osapeittoja.

Ensimmäinen haaste oli luonnollisesti löytää julkisen avoimen datan arkisto; jokin taho, jolla olisi intressejä tallentaa vallitsevaan hetkeen sidonnaisia tiedotteita suurissa määrin. Tällaiseksi osoittautui yhdysvaltalaisen Iowa Environmental Mesonet'in (IEM) ylläpitämä kanta. Suuremmaksi ja käytännön ongelmaksi tunnistettiin METAR-sanomien epäsäännöllinen rakenne. Säähavaintotietojen etsiminen datamassasta vaati siten erityisesti siihen soveltuvien työkalujen käyttöä.

Ratkaisuna haluttiin tutkia tyypillisiä UNIX-työkaluja, kuten nimenomaan tekstitiedostojen käsittelyyn suunniteltua AWK-kieltä ja komentorivin perustyökaluja kuten *grep*, *cat*, *sort*, ja *tr* sekä esimerkiksi BASH-skriptausta, jolla aiemmin mainituista työkaluista voitaisiin niputtaa oikeita helppokäyttöisiä prosesseja. Havaittiin, että pienellä harjoittelulla työkalut sopivat erityisen hyvin epäsäännöllisen datan suodattamiseen ja käsittelyyn.

## 2. METAR-SANOMA

### 2.1. Mikä on METAR-sanoma?

METAR on lyhenne sanoista METeorological Airport Report, jonka suomenkielinen vastine on lentosäähavaintosanoma. Lentosäähavaintosanomien sisällön, muodon ja frekvenssin määrittelee kansainvälinen siviili-ilmailujärjestö ICAO (International Civil Aviation Organization). Suomessa METAR -sanomia tuotetaan kahdesti tunnissa. [1]

METAR-sanoma perustuu pintasäähavaintoihin ja on luonteeltaan pistemäinen ja hyvin paikallinen. Yksittäinen sanoma on tulos ajallisesta keskiarvoituksesta ja kuvaa siten ennemminkin pidemmän ajan ilmiötä kuin suoraa mittauslukemaa yksittäiseltä anturilta tietyllä ajan hetkellä. Juuri tämän hetken sää voi siten poiketa merkittävästikin juuri puolituntia sitten lähetetyn METAR-sanoman informaatiosta. Hyvin yllättävien ja äkillisten säätilan muutosten yhteydessä voidaan lähettää ylimääräinen METAR-sanoma. [1]

Kuvatusta syistä johtuen METAR-sanomien pohjalta ei tulisi tehdä kovin laajoja tulkintoja alueellisesta säästä vaan käsitellä sitä lennonsuunnittelullisiin tarkoituksiin soveltuvana. [1] Tässä työssä METAR-sanomia käytetään peruutuspeiliin katsoen, todeten: ”Että näin on tapahtunut ja tällainen pilvisuus on vallinnut tämän puolituntisen ajan.”

### 2.2. Muista sanomatyypeistä

Siinä missä METAR-sanoma kuuluu havaintoihin (OBS, observation), lentokentät tarjoavat myös TAF-sanomia, jotka ovat ennusteita (FCST, forecast). TAF-tiedotteita käytetään lennonsuunnitteluun. Suomessa tällaisia ennusteita tuotetaan kolmen tunnin välein. Merkittävien muutosten tapauksessa voidaan tuottaa erityinen TAF AMD (amendment) -sanoma. TAF-sanomat noudattavat samaa muotoa ja koodausta kuin METAR-tiedote – vain etuliite on eri. [1][2]

Ennusteet eivät luonnollisesti ole mielekkäitä tämän työn kannalta ja siten niiden tarkempi tarkastelu sivuutetaan.

Ajastettujen METAR -sanomien lisäksi ”operatiivisesti merkittävästi muuttuneista” sääoloista tuotetaan erityisiä, ajastamattomia SPECI-havaintoja (SPECIAL). Tarkemmat parametriset rajat on esitetty mm. Ilmatieteenlaitoksen Lentosääpalvelut Suomessa -julkaisussa. Suomessa SPECI-havainnot eivät ole julkisia. Tätä perustellaan kaksi kertaa tunnissa annettavilla METAR-tiedotteilla. SPECI-havainto noudattaa samaa koodausta kuin METAR-viesti, jonka lisäksi se saattaa sisältää selkokielistä, koodaamatonta informaatiota. Viestit erottaa toisistaan parhaiten alun tunnisteesta METAR tai SPECI. [1][2]

Edellä mainittujen lisäksi lentokentät tuottavat MET REPORT -tiedotteita, joita julkaistaan samalla hetkellä kuin METAR-sanoma. MET REPORT- tiedotteen tulokset esimerkiksi tuulennopeudelle saattavat poiketa tarkemmin säädellyistä METAR-sanomista ja niiden tarkoitus on paikallissanomana tukea paikallista operatiivista toimintaa. [1][2] **Error! Reference source not found.**

Tässä työssä keskitytään METAR-sanomiin ja niiden sisältämään pilvisyysdataan.

### 2.3. METAR-tiedotteen rakenne

METAR-sanoma noudattaa muotoa:

**METAR\_CCCC\_YYGGggZ AUTO/COR\_dddff(f)Gf<sub>m</sub>f<sub>m</sub>(f<sub>m</sub>)KT\_d<sub>n</sub>d<sub>n</sub>d<sub>n</sub>Vd<sub>x</sub>d<sub>x</sub>d<sub>x</sub>  
\_VVVVVSM\_[RD<sub>R</sub>DR/V<sub>R</sub>V<sub>R</sub>V<sub>R</sub>V<sub>R</sub>FT/RD<sub>R</sub>DR/V<sub>N</sub>V<sub>N</sub>V<sub>N</sub>V<sub>N</sub>VV<sub>X</sub>V<sub>X</sub>V<sub>X</sub>V<sub>X</sub>FT]\_w'w'  
'\_[N<sub>s</sub>N<sub>s</sub>N<sub>s</sub>h<sub>s</sub>h<sub>s</sub>h<sub>s</sub>orVVh<sub>s</sub>h<sub>s</sub>h<sub>s</sub>orSKC/CLR]\_T'T'/T'dT'd\_AP<sub>H</sub>P<sub>H</sub>P<sub>H</sub>P<sub>H</sub>\_RMK**, jossa

- \_ -merkki kuvaa välilyöntiä,
- CCCC kertoo lentokentän yksilöllisen tunnisteiden (esimerkiksi EFOU),
- GGggZ mittausajankohdan,
- AUTO/COR onko informaatio täysin automaattista tai korjattua,
- dddff(f)Gf<sub>m</sub>f<sub>m</sub>(f<sub>m</sub>)KT\_d<sub>n</sub>d<sub>n</sub>d<sub>n</sub>Vd<sub>x</sub>d<sub>x</sub>d<sub>x</sub> tuuli-informaation,
- VVVVVSM näkyvyyden,
- RD<sub>R</sub>DR/V<sub>R</sub>V<sub>R</sub>V<sub>R</sub>V<sub>R</sub>FT tai RD<sub>R</sub>DR/V<sub>N</sub>V<sub>N</sub>V<sub>N</sub>V<sub>N</sub>VV<sub>X</sub>V<sub>X</sub>V<sub>X</sub>V<sub>X</sub>FT kiitoratanäkyvyyden,
- w'w' tämän hetkisen sään,
- N<sub>s</sub>N<sub>s</sub>N<sub>s</sub>h<sub>s</sub>h<sub>s</sub>h<sub>s</sub> tai VVh<sub>s</sub>h<sub>s</sub>h<sub>s</sub> tai SKC/CLR taivaan ja pilvisyyden tilan,
- T'T'/T'dT'd lämpötilan ja kastepisteen sekä
- AP<sub>H</sub>P<sub>H</sub>P<sub>H</sub>P<sub>H</sub> ilmanpaineen merenpinnan tasolla. [3]

Perusrakenteen lisäksi tiedotteen lopussa voi olla **RMK** (Remarks section of report) eli huomautukset -osio, johon voidaan lisätä automatisoitua tai manuaalista koodaamatonta informaatiota tekstimuodossa. [3] Ilmatieteenlaitoksen mukaan [1] Suomessa tällainen koodaamaton lisä esiintyy lähinnä SPECI-sanomissa ja mahdollisesti MET REPORT -tiedotteissa.

Seuraavassa on esitetty käytännön esimerkkinä Oulun lentoaseman METAR-tiedote sunnuntiaamulta 18.11.2018, klo 10:20:

**EFOU 180820Z AUTO 01007KT CAVOK 01/M03 Q1043.**

Suomen kaikki lentokentät ovat EF-alkuisia. OU-lopuke kertoo tarkennuksena, että kyseessä on Suomen kentistä Oulun lentokenttä, eli EFOU. Helsinki-Vantaan lentokentän tunnistaa lyhenteestä EFHK ja esimerkiksi Vaasan lentokentän lyhenteestä EFVA.

**GGggZ** -osiosta nähdään, että tiedote on kuluvan kuukauden 18. päivästä (marraskuu), kello 8:20 standardiaikaa [1][3]. Standardiaikaan tulee lisätä kaksi tuntia, sillä suomi noudattaa aikavyöhykettä GMT+2. Tässä kohdin on myös hyvä havaita, että METAR-sanoma ei itsessään sisällä tietoa kuukaudesta tai kuluva vuodesta.

Z-kirjain osion lopussa indikoi Zulu-aikaa, eli standardiaikaa. Tämä mitä ilmeisimmin juontaa ns. NATO:n (North Atlantic Treaty Organization) foneettisista aakkosista, jotka on sittemmin standardoitu kansainvälisen siviili-ilmailujärjestön ICAO:n (International Civil Aviation Organization) toimesta. Faktatietoa asiasta ei kuitenkaan ole saatavilla.

**AUTO** -osiossa on määritelty, että lähete on automaattisista mittauksista kerätty ja automaattisesti tuotettu. Suomessa ainoastaan Helsinki-Vantaalla tehdään lisäksi



manuaalisia säähavaintoja [1]. Silti kaikki työssä käsitellyt sanomat eivät sisältäneet AUTO-tunnistetta.

Hankalan näköisesti koodatun **dddff(f)Gf<sub>m</sub>f<sub>m</sub>(f<sub>m</sub>)KT<sub>n</sub>d<sub>n</sub>d<sub>n</sub>Vd<sub>x</sub>d<sub>x</sub>d<sub>x</sub>** vastine esimerkiviestissä on yksinkertaisesti 01007KT. ”010” (**ddd**) tarkoittaa, että tuulen suunta on 10 astetta pohjoisesta (koillista kohden) ja että tuulen nopeus on 7 solmua, eli 3,6 m/s (**ff(f)**) [3]. Suluissa oleva f-kirjain viittaa siihen, että tuulen nopeus voidaan ilmaista myös kolmella numerolla. 100 solmua vastaa noin 51,4 m/s. G-kirjaimen jälkeinen osa sisältää puuskatietoja.

Ilmatieteenlaitoksen mukaan Suomen keskituulen nopeusennätys on 32,5 m/s, joka mitattiin 2.1.2019 Kökarin Bogskärissä. Sitä edeltävä ennätys oli 31 m/s vuodelta 1971 ja mitattu Valassaarilta. Suurimmat tuulennopeudet on Suomessa mitattu tuntureiden huipuilta, n. 40 m/s ja puuskissakin n. 50 m/s. [4] Kotimaisilla kentillä riittää siten kahden numeron ilmaisu sekä keskituulen että puuskien osalta.

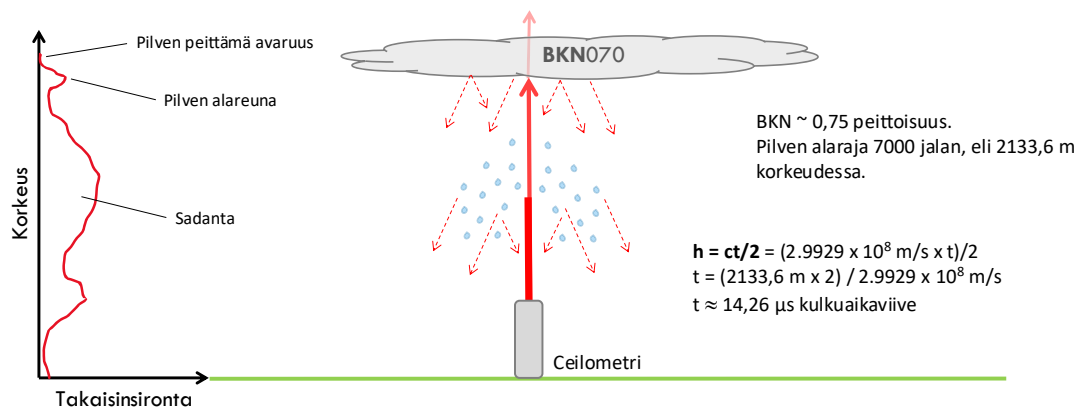
Lisähuomiona tulee todeta, että **ddd** -osio voi olla ilmoitettu myös kirjainlyhenteellä **VRB** (variable), jolla viestitään, että tuulen suunta on vaihteleva ja että tuulen nopeus on korkeintaan 6 solmua eli noin 3,1 m/s. Jos tuulennopeus vaihtelee vähintään 180 astetta, lyhennettä voidaan käyttää suuremmillakin nopeuksilla. [1]

Tuulen nopeuden ollessa suurempi kuin 6 solmua ja suunnan vaihdella otetaan käyttöön **KT**:n jälkeinen **d<sub>n</sub>d<sub>n</sub>d<sub>n</sub>Vd<sub>x</sub>d<sub>x</sub>d<sub>x</sub>** -osio. [3]

**CAVOK** (Ceiling and Visibility OK) tarkoittaa yksinkertaisesti näkyvyys on hyvä ja esteetön. Vastaavan tiedon olisi antanut myös **SKC** (Sky Clear). Ilmaisut **FEW** (few), **SCT** (scattered), **BKN** (broken) ja **OVC** (overcast) edustavat eri asteista pilvipeittoa ja niitä käsitellään tarkemmin seuraavassa. Esitellyistä parametreista tässä työssä keskitytään siis näihin.

## 2.4. Taivaan peittoisuus

Pilvipeitteisyyden mittaamiseen käytetään ceilometriä, eli pilvenkorkeusmittaria [1][2]. Mittaus perustuu optiseen tutkaan (LIDAR, light detection and ranging), joka mittaa pulssitetun laserin kuluaikaviivettä. Kuva 1 esittää yksinkertaistetun pilvenkorkeusmittarin toimintaperiaatteen. Vasemmalla käyrässä näkyy alempana sateen tuottamaa sirontaa. 7000 jalan korkeudessa sijaitsee noin 75 % pilvipeitteen alaraja. Kaavan 1 mukaisesti 7000 jalan, eli 2133,6 m korkeutta vastaava kuluaikaviive olisi noin 14,26 μs. [5]



Kuva 1. Havainnekuva pilvenkorkeusmittarin toimintaperiaatteesta.

$$t = \frac{2h}{c}, \text{ jossa } c \text{ on valonnopeus.} \quad (1)$$

Tyypillinen lentoasemalta löytyvä pilvenkorkeusmittari voisi olla esimerkiksi kotimaisen Vaisalan CL31, joka perustuu InGaAs -diodiin, jonka aallonpituus on 910 nm. Mittausintervalli on 2-3 s ja sillä voidaan erottaa kolme eri pilvikerrosta 7.6 km korkeuteen saakka. [5] Koska yksittäinen ceilometri mittaa varsin pistemäistä kohtaa ja juuri yläpuolellaan olevia pilviä, realistisemman mittaustuloksen saamiseksi tulos muodostetaan noin puolen tunnin ajanjaksolta. [1]

Tämän työn kannalta pilvenkorkeus ei ole oleellinen, ainoastaan taivaan kokonaispeittoisuus. Täten, tarkempi analyysi kerroksellisuudesta sivuutetaan ja keskitytään kokonaispilvipeittoon.

Seuraavassa taulukossa (Taulukko 1) on kuvattu METAR-tiedotteen taivaan peittoisuusasteelle antamia tunnisteita. Näitä ovat ensimmäisen sarakkeen viisi ensin mainittua. Peittoisuusaste on kuvattu kolmikirjaimisella lyhenteellä. Osa lyhenteistä on yksiselitteisiä, kuten pilvetön taivas (SKC, Sky Clear) tai pilvinen (OVC, Overcast), joka on määritelty täysin pilvipeitteiseksi. Tältä väliltä löytyy kolme peittoisuusluokkaa FEW (Few), SCT (Scattered) ja BKN (Broken). Ne kattavat kukin tietyt paloittain määritellyt peittoisuusalueet. [1][3] Lisäksi tunnetaan nykyisin myös muita ilmaisuja kuten ”operatiivisesti ei merkittäviä pilviä”, eli NSC (No Significant Cloud) sekä NCD (No Clouds Detected) [1]. Viimeksi mainittuja ilmaisuja ei ilmennyt tässä työssä käytetyssä datassa.

Taulukko 1. Pilvisyys ja taivaan peittoisuusaste.

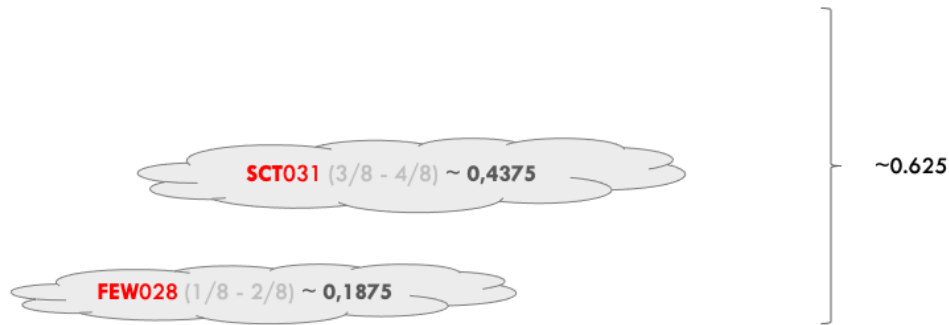
| <b>Tunnisteen etuosa</b> | <b>Termi</b>                 | <b>Peittoisuusaste</b> | <b>Kerroin</b> |
|--------------------------|------------------------------|------------------------|----------------|
| SKC                      | Sky Clear                    | 0/8                    | 1              |
| FEW                      | Few                          | 1/8 – 2/8              | 0,8125         |
| SCT                      | Scattered                    | 3/8 – 4/8              | 0,5625         |
| BKN                      | Broken                       | 5/8 – 7/8              | 0,25           |
| OVC                      | Overcast                     | 8/8                    | 0              |
| VV                       | Vertical<br>Visibility       | -                      | 0              |
| CAVOK                    | Ceiling and<br>Visibility OK | -                      | 1              |

Lisäksi taivaan peittoa ja näkyvyyttä voidaan kuvata vertikaalinäkyvyydellä VV (Vertical Visibility) sekä lyhenteellä CAVOK (Ceiling and Visibility OK). [3] Ensimmäinen mainittu on sikäli sukua OVC-tunnisteelle, että molemmissa taivaan katsotaan olevan täysin pilvinen. CAVOK on vastaavasti sukua SKC-tunnisteelle. Kummassakaan määritelmässä ei oteta kantaa taivaan peiton kerroksellisuuteen. Tämän työn kannalta SKC and CAVOK edustavat esteetöntä taivasta ja OCV sekä VV täysin pilvipeitteistä taivasta.

METAR-tiedotteessa pilvisyysastetta ilmaisevien lyhenteiden perässä on aina kolme numeroa, jotka ilmaisevat pilven alarajan korkeuden satoina jalkoina. [1][3] Pilvien korkeus ei ole kiinnostavaa tämän työn kannalta. Varsinaisessa laskennassa

korkeusinformaatio sivuutetaan, mutta selkeyden takia se huomioidaan osassa esimerkeistä.

EFOU 2020-06-06 08:50 EFOU 060850Z AUTO 17009KT 150V210 9999 FEW028 SCT031 19/11 Q1000



Kuva 2. Esimerkki kokonaispilvipeittoisuuden laskennasta.

Laskennan kannalta arviotyypinen ilmaisu, esimerkiksi FEW (1/8-2/8 peittoisuus) on ongelmallinen. Tästä johtuen kertoimiksi on laskennassa määritetty arvoalueen keskiarvot. Taulukko 1 tiivistää nämä kertoimet.

Kuva 2 esittää käytännön esimerkin (6.6.2020, klo 8:50) kokonaispilvipeittoisuuden laskemisesta yksittäisestä METAR-tiedotteesta. Tiedotteen mukaisesti on tunnistettu kaksi selkeästi erillistä pilvikerrosta: alempi reilun 853 m (2800 jalan) korkeudessa sekä ylempi noin 945 metrin (3100 jalan) korkeudessa. Alemman kerrostuman peittoaste laskennassa on 0,1875 ja ylemmän kerrostuman 0,4375. Siten kokonaispeittoisuudeksi saadaan 0,625. Toisin sanoen, 37,5 % valosta läpäisee kerrokset. Käytännössä on mahdollista, että yhteen laskettu peitteisyys on lukuarvoltaan yli yhden. Laskennassa kaikki numeroa yksi suuremmat lukemat oletetaan täydeksi pilvipeitteisyydeksi, eli numeroksi yksi.

### 3. AINEISTO

#### 3.1. METAR-tiedotteet

Tämän työn perustana toimi Iowa Environmental Mesonet'in (IEM) ylläpitämä METAR-tiedotepankki [6], josta oli ladattavissa kaikki Oulun lentoaseman METAR-tiedotteet vuoden 2010 alusta lähtien. Työssä käytettiin raakamuotoisia METAR-tiedotteita kokonaisen vuosikymmenen 2010–2020 ajalta. Kymmenen vuoden tiedotteet mahtuivat 191422 riviin ja vajaan 16 Mt tekstitiedostoon, jonka avaaminen esimerkiksi tyypillisellä toimisto-ohjelmistolla (esimerkiksi Microsoft Excel) osoittautui käytännössä mahdottomaksi. Varsinaista hyötyä toimisto-ohjelmistosta ei METAR-viestien tulkintaan olisi ollutkaan; viesti ei ole määrämittainen eikä rakenteeltaan vahvasti vakioitu.

| station | valid            | metar   |
|---------|------------------|---|
| EFOU    | 2010-01-01 00:20 | EFOU 010020Z AUTO 05009KT 020V100 9999 -SN OVC025 M10/M12 Q1012                   |
| EFOU    | 2010-01-01 00:50 | EFOU 010050Z AUTO 05011KT 6000 -SN OVC026 M10/M12 Q1012                           |
| EFOU    | 2010-01-01 01:20 | EFOU 010120Z AUTO 05009KT 9999 -SN OVC030 M10/M12 Q1013                           |
| EFOU    | 2010-01-01 01:50 | EFOU 010150Z AUTO 05010KT 9999 -SN BKN031 OVC066 M10/M13 Q1013                    |
| EFOU    | 2010-01-01 02:20 | EFOU 010220Z AUTO 05011KT 9999 -SN FEW027 BKN068 M10/M13 Q1013                    |
| EFOU    | 2010-01-01 02:50 | EFOU 010250Z AUTO 05011KT 9999 -SN BKN032 BKN068 M10/M13 Q1014                    |
| EFOU    | 2010-01-01 03:20 | EFOU 010320Z AUTO 05011KT 9999 -SN SCT033 OVC065 M10/M13 Q1014                    |
| EFOU    | 2010-01-01 03:31 | EFOU 010331Z 06010KT 9999 -SN FEW035 BKN060 M11/M13 Q1014                         |
| EFOU    | 2010-01-01 03:50 | EFOU 010350Z 06010KT 9999 -SN FEW037 BKN060 M11/M14 Q1014 12390154                |
| EFOU    | 2010-01-01 04:20 | EFOU 010420Z 05009KT 9999 -SN FEW030 SCT045 BKN070 M11/M14 Q1014 12390154         |
| EFOU    | 2010-01-01 04:50 | EFOU 010450Z 05007KT 020V090 9999 -SN FEW030 BKN045 BKN070 M11/M14 Q1015 12390154 |
| EFOU    | 2010-01-01 05:20 | EFOU 010520Z 04007KT 010V080 9999 -SN FEW030 BKN040 M11/M14 Q1015 12390154        |
| EFOU    | 2010-01-01 05:50 | EFOU 010550Z 04009KT 9999 -SN FEW030 BKN040 M11/M14 Q1015 12390154                |
| EFOU    | 2010-01-01 06:20 | EFOU 010620Z 05008KT 010V070 9999 -SN FEW030 BKN040 M12/M15 Q1016 12390154        |
| EFOU    | 2010-01-01 06:50 | EFOU 010650Z 05007KT 9999 -SN FEW030 BKN040 M12/M15 Q1016 12390154                |
| EFOU    | 2010-01-01 07:20 | EFOU 010720Z 05006KT 360V090 9999 -SN BKN038 M12/M15 Q1016 12390154               |
| EFOU    | 2010-01-01 07:50 | EFOU 010750Z 05006KT 010V080 9999 BKN038 M12/M15 Q1017 12390154                   |
| EFOU    | 2010-01-01 08:20 | EFOU 010820Z 03006KT 9999 -SN BKN037 M12/M14 Q1017 12390154                       |
| EFOU    | 2010-01-01 08:50 | EFOU 010850Z 05008KT 5000 -SN FEW030 OVC037 M12/M14 Q1017 12390154                |
| EFOU    | 2010-01-01 09:20 | EFOU 010920Z 04008KT 7000 -SN FEW030 BKN037 BKN120 M12/M14 Q1018 12390154         |
| EFOU    | 2010-01-01 09:50 | EFOU 010950Z 05008KT 020V080 7000 -SN FEW018 SCT030 BKN120 M12/M14 Q1018 12390154 |
| EFOU    | 2010-01-01 10:20 | EFOU 011020Z 05009KT 9999 -IC FEW030 SCT200 M12/M15 Q1018 12390154                |
| EFOU    | 2010-01-01 10:50 | EFOU 011050Z 05009KT 9999 FEW035 M12/M15 Q1018 12390154                           |
| EFOU    | 2010-01-01 11:20 | EFOU 011120Z 04009KT 9999 FEW035 M13/M15 Q1018 12390154                           |
| EFOU    | 2010-01-01 11:50 | EFOU 011150Z 03008KT 9999 FEW035 M13/M15 Q1018 12390171                           |
| EFOU    | 2010-01-01 12:20 | EFOU 011220Z 03008KT 9999 FEW035 M13/M16 Q1019 12390171                           |
| EFOU    | 2010-01-01 12:50 | EFOU 011250Z 03008KT 9999 FEW035 M14/M16 Q1019 12390171                           |
| EFOU    | 2010-01-01 13:20 | EFOU 011320Z 03008KT 9999 FEW040 M14/M16 Q1019 12390171                           |
| EFOU    | 2010-01-01 13:50 | EFOU 011350Z 04009KT CAVOK M14/M17 Q1019 12390171                                 |
| EFOU    | 2010-01-01 14:20 | EFOU 011420Z 04008KT 350V060 CAVOK M15/M17 Q1020 12390171                         |
| EFOU    | 2010-01-01 14:50 | EFOU 011450Z 04009KT CAVOK M15/M18 Q1020 12390171                                 |
| EFOU    | 2010-01-01 15:20 | EFOU 011520Z 04008KT CAVOK M15/M18 Q1020 12390171                                 |
| EFOU    | 2010-01-01 15:50 | EFOU 011550Z 04008KT 010V070 CAVOK M15/M18 Q1020 12390186                         |
| EFOU    | 2010-01-01 16:20 | EFOU 011620Z 05010KT CAVOK M15/M18 Q1020 12390186                                 |
| EFOU    | 2010-01-01 16:50 | EFOU 011650Z 06010KT CAVOK M15/M18 Q1021 12390186                                 |
| EFOU    | 2010-01-01 17:20 | EFOU 011720Z 06009KT 020V090 CAVOK M15/M17 Q1021 12390186                         |
| EFOU    | 2010-01-01 17:50 | EFOU 011750Z 05009KT CAVOK M15/M18 Q1021 12390186                                 |
| EFOU    | 2010-01-01 18:20 | EFOU 011820Z 06008KT 9999 BKN037 M15/M18 Q1022 12390186                           |
| EFOU    | 2010-01-01 18:50 | EFOU 011850Z 06008KT 9999 BKN037 M15/M18 Q1022 12390186                           |
| ...     |                  |   |

Kuva 3. METAR-aineiston 20 ensimmäistä tiedotetta (EFOU.txt).

Kuva 3 esittää otteen käytetystä METAR-aineistosta. Listauksen metar-sarake sisältää alkuperäiset tiedotteet. Virallinen tiedote ei sisällä itsessään aikaleimaa, joten IEM on lisännyt kantaansa erillisen valid-sarakkeen, josta käy ilmi tiedotteen julkaisuajankohta. Tuntemattomasta syystä aineisto sisälsi myös aseman tunnusarakan, vaikka tiedotteesta tämä asia käy yksiselitteisesti ilmi. Aikaleimoja hyötykäytettiin itse työssä sekä kohdistamaan eri datajoukot keskenään että itseisarvoisesti sijoittamaan pilvipeteinformaatio oikeaan ajanhetkeen.

METAR-aineisto ladattiin tietokannasta 1.4.2021. Aineiston käytössä huomioitiin, että sen aikaleimat olivat nk. Zulu-aikaa, eli ilmoitettu GMT+0 -ajassa. Näin ollen kuvan neljä ensimmäistä tiedotetta kuuluivat vuoden 2009 puolelle ja vastaavasti neljä ensimmäistä vuoden 2021 tiedotetta olivat todellisuudessa vuoden 2020 neljä viimeistä. Tarvittava siirtymä huomioitiin käsittelyssä ja raaka-aineiston pituudessa. Myöhemmin aineiston käsittelyn yhteydessä havaittiin, että käytännön säädatalle tyypillisesti useita tiedotteita puuttui välistä. Tämän ongelman ratkaisuun palataan seuraavissa luvuissa tarkemmin.

### 3.2. Aurinkodata sekä korjaava aikasarja METAR-tiedoille

Tämä työ keskittyi METAR-sanomien pilvisyystiedon laskemiseen. Koska käytetyissä METAR-tiedotteissa havaittiin tarkemman käsittelyn yhteydessä ylimääräisiä sanomia sekä useita välistä puuttuvia tiedotteita ja koska pilvisyystietojen mahdollinen jatkohyödyntäminen – joka tässä tapauksessa jää tämän kandidaatin työn ulkopuolelle – vaatii useiden eri datajoukkojen yhdistämistä, päätettiin käyttää korjausapuna ulkoista oikeellisen aikasarjan sisältävää datajoukkoa. Ilman edellä mainittua korjausta eri datajoukkojen yhdistäminen ei onnistuisi; aikaleimat eivät täsmäisi ja tapahtumat sijoittuisivat ajallisesti väriin kohtiin.

Varsin hyödylliseksi tähän tarkoitukseen osoittautui amerikkalaisen National Renewable Energy Laboratory'n (NREL) [7] SOLPOS-laskuri, jonka avulla voitiin tuottaa halutuissa koordinaateissa datasarja ideaaliselle, Maan ilmakehän ulkopuoliselle aurinkoenergialle. Aurinkodata voitiin oikeellisine aikasarjoineen tuottaa Oulun lentoaseman koordinaateissa. Tätä voidaan hyödyntää tulevilla töissä.

```
Date,Time,Elevation (refracted),ETR global,ETR direct
```

```
...
```

```
1/1/2010,9:00:00,-6.0225,0.0000,0.0000
1/1/2010,9:10:00,-5.2722,0.0000,0.0000
1/1/2010,9:20:00,-4.5493,0.0000,0.0000
1/1/2010,9:30:00,-3.8542,0.0000,0.0000
1/1/2010,9:40:00,-3.1867,0.0000,0.0000
1/1/2010,9:50:00,-2.5453,0.0000,0.0000
1/1/2010,10:00:00,-1.9266,0.0000,0.0000
1/1/2010,10:10:00,-1.3208,0.0000,0.0000
1/1/2010,10:20:00,-0.7000,0.0000,0.0000
1/1/2010,10:30:00,0.0174,0.4308,1414.9135
1/1/2010,10:40:00,0.3863,9.5405,1414.9135
1/1/2010,10:50:00,0.7330,18.1010,1414.9135
1/1/2010,11:00:00,1.0529,25.9988,1414.9135
1/1/2010,11:10:00,1.3424,33.1476,1414.9135
1/1/2010,11:20:00,1.5989,39.4804,1414.9135
1/1/2010,11:30:00,1.8203,44.9450,1414.9135
1/1/2010,11:40:00,2.0050,49.5032,1414.9135
1/1/2010,11:50:00,2.1518,53.1252,1414.9135
1/1/2010,12:00:00,2.2598,55.7900,1414.9135
1/1/2010,12:10:00,2.3284,57.4829,1414.9135
1/1/2010,12:20:00,2.3572,58.1953,1414.9135
1/1/2010,12:30:00,2.3462,57.9235,1414.9135
1/1/2010,12:40:00,2.2954,56.6691,1414.9135
1/1/2010,12:50:00,2.2050,54.4379,1414.9135
1/1/2010,13:00:00,2.0755,51.2428,1414.9135
1/1/2010,13:10:00,1.9077,47.1008,1414.9135
1/1/2010,13:20:00,1.7026,42.0385,1414.9135
1/1/2010,13:30:00,1.4616,36.0893,1414.9135
1/1/2010,13:40:00,1.1865,29.2994,1414.9135
1/1/2010,13:50:00,0.8799,21.7276,1414.9135
1/1/2010,14:00:00,0.5448,13.4531,1414.9135
1/1/2010,14:10:00,0.1853,4.5750,1414.9135
1/1/2010,14:20:00,-0.3942,0.0000,0.0000
1/1/2010,14:30:00,-1.0474,0.0000,0.0000
1/1/2010,14:40:00,-1.6541,0.0000,0.0000
1/1/2010,14:50:00,-2.2651,0.0000,0.0000
1/1/2010,15:00:00,-2.8956,0.0000,0.0000
1/1/2010,15:10:00,-3.5510,0.0000,0.0000
...
```

Kuva 4. Näyte SOLPOS-aineiston ensimmäisestä päivästä (solar\_2010-2013.txt).

Kuva 4 esittää näytteen SOLPOS-aineiston ensimmäiseltä päivältä (rivit 56-93). Voidaan helposti todeta, että aurinko on noussut vasta noin klo 10:30 aikoihin ja painunut mailleen jo klo 14:20. Nähdään myös, että alkuperäinen aineisto on kymmenen minuutin välein tuotettua. Johtuen Oulun lentoaseman kaksikymmentä yli ja kymmentä vaille tasatunnin annettujen tiedotteiden, jouduttiin valitsemaan laskurille sellainen näytteistys, joka tuottaisi datapisteen samoille aikaleimoille. Neljä ylimääräistä datapistettä per tunti oli helposti poistettavissa datan käsittelyn yhteydessä.

SOLPOS-aineiston latauksen yhteydessä havaittiin mielenkiintoinen rajoite; toisin kuin Iowa Environmental Mesonet'in METAR-aineiston tapauksessa, NREL'in laskurista täytyi kokonaisen vuosikymmenen aineisto ladata useassa erässä (2010-2013, 2014-2019 sekä 2020). Nämä yhdistettiin yhdeksi käsiteltäväksi tiedostoksi käyttäen AWK-kieltä sekä UNIX-komentorivin käskyjä (ks. skripti *prepareSolar.sh*, Liite 2).

## 4. DATAN KÄSITTELY JA POHDINTA

### 4.1. Käsittelyn vaiheet

Käytännön datan käsittely jaettiin seuraaviin osiin:

1. METAR-tiedotteiden valmistelu ja esikäsittely (Liite 1, *prepareMETAR.sh*),
2. aurinkodatan valmistelu (Liite 2, *prepareSolar.sh*),
3. METAR-aikasarjan korjaus (Liite 3, *patchMETARemptyDates.sh*),
4. Tyhjien METAR-rivien täyttö (Liite 4, *fillEFOUgaps.sh*) sekä
5. METAR- ja aurinkodatojen yhdistäminen (Kuva 13, Kuva 14).

### 4.2. METAR-tiedotteiden valmistelu ja esikäsittely

METAR-datan valmistelu suoritettiin *prepareMETAR.sh* -skriptillä, jonka listaus on esitetty liitteessä 1. Pilvisyys on koodattu METAR-tiedotteisiin kirjainlyhenteillä, jotka ovat suoran laskennan kannalta käyttökelvoton muoto. Lisäksi pilvisyyden ilmaisemisessa on päällekkäisyyttä, jota on käsitelty tarkemmin luvussa 2.4 Taivaan peittoisuus. Valmistelussa etsittiin rivipohjaisesti vuorollaan kaikkia seitsemää kirjainlyhennettä ja muodostettiin jokaisen rivin eteen niitä vastaavat numerokertoimet (omalle seitsemälle sarakkeelleen). Kokonaispilvipeite määritettiin laskemalla yhteen kaikki seitsemän osakerrosta (*sky\_coverage.awk*).

Myös METAR-tiedotesarjan puutteellisuuksia tuli korjata; Oulun lentoaseman METAR-tiedotteet lähetetään kahdesti tunnissa, kaksikymmentä yli ja kymmentä vaille tasatunnin. Jos säähavainnoissa tapahtuu merkittävä muutos, lähetetään uusi erillinen tiedote. Näytteistykseen tasaisuuden kannalta tämä oli ongelmallista, sillä eri päivämäärillä saattoi olla poikkeava määrä tiedotteita. Lisäksi käytetyssä datassa ilmeni yksittäisiä tiedotteita kahdesti tai jopa useampaankin kertaan peräkkäin. Myös tällöin yhdelle päivämäärälle kertyi poikkeava määrä tiedotteita.

Koska näytteistys täytyi täsmentää, aluksi skriptissä suodatettiin pois kaikki muut aikaleimat paitsi :20- ja :50 -päätteiset. Suuressa mittakaavassa virhe jäi arviolta hyvin pieneksi, vaikka näin saatettiinkin menettää noin toistakymmentä tiedotetta vuositasolla (ja viestien pilvisyyden painotettu keskiarvo olisi saattanut olla parempi arvio tilanteesta).

Kuva 5 esittää komentorivin tulostuksen edellä kuvattujen vaiheiden mukaisesti. Ylimääräiset tiedotemonikerrat poistettiin käyttäen AWK-kielen riveittäisen käsittelyn ominaisuutta; etsimällä päivämäärien epäjatkuvuuskohtia ja poistamalla nämä rivit. Tällä tavoin useammat kuin kaksinkertaiset viestit jäävät ilmaistumatta, joten ominaisuutta päätettiin käyttää iteroiden. Tämä näkyy kuvan listauksessa.

Tämän jälkeen tuli vielä huomioida METAR-tiedotteiden GMT-0-aikasadonnaisuus. Jotta käsiteltyä dataa voitaisiin mahdollisesti jatkoehdyntää, korjattiin aika GMT+2 mukaiseksi, ts., tehtiin datalle neljän näytteen siirto.

```

Validating data...
Data validated in 0 seconds.

Processing data...
Sky coverage processed in 7 seconds.

Processing METAR data for duplicates and multiple entries...
37 multiple lines found...
2 unprocessed multiple lines remains...
0 unprocessed multiple lines remains...

Fixing UTC-0 timezone to UTC+2.
Timezone fixed.
Recreating original data and time format.

Data processed and results written in 11 seconds.

user@users-Mac-mini Kandi %

```

Kuva 5. prepareMETAR.sh BASH-skriptin komentorivin tulostus.

### 4.3. Aurinkodata ja METAR-datan puutteet

METAR-tiedotteille on luonteenomaista, että ”viimeisin on aina voimassa”. Aiempiin tiedotteisiin ei palata, vaan ainoastaan viimeisin on oleellinen ja ”paras näkemys tilanteesta”. Tästä ja säähavaintodatan luonteesta johtuen joitain havaintoja on saattanut jäädä lähettämättä (esimerkiksi säälaitteiden huoltokatkokset tai järjestelmien vikatilat) ja siten kirjaamatta tietokantaan. Aiemmin poistettiin ylimääräiset viestimonikerrat käyttäen hyväksi viisitasoista laskuria (vuosi, kuukausi, päivä, tunnit ja minuutit). Tämä AWK-kielen ominaisuus ei ota kuitenkaan huomioon välistä puuttuvia yksittäisiä viestejä vaan tarkastelee ainoastaan, että seuraava luku on suurempi kuin tämänhetkinen.

Tätä ongelmaa voitiin lähestyä kahta reittiä, joko testata METAR-dataa suoraan oikeellisia päivämääriä vasten tai käyttäen avuksi toista oikeellista datajoukkoa, josta oikeelliset aikaleimat löytyvät jo valmiiksi. Koska välistä puuttuvat METAR-tiedotteet ilmenivät kahden eri datajoukon yhdistämisen yhteydessä (kuten monesti mittausdatan käsittelyn yhteydessä käy), päätettiin puuttuvat kohdat tilkitä hyötykäyttämällä aurinkodatan aikaleimoja. Aurinkodataa käytettiin tässä työssä vain esimerkinomaisesti ja jatkosovellukset jäävät myöhemmissä töissä käsiteltäviksi.

Aurinkodatana käytettiin Yhdysvaltain kansallisen uusiutuvan energian laboratorion (NREL, National Renewable Energy Laboratory) SOLPOS -laskuria, jolla teoreettisen aurinkoisuuden voi laskea tietylle koordinaattipisteelle maapallolla – tässä tapauksessa Oulun lentoaseman koordinaateihin. Tämän työn kannalta käyttökelpoisin työkalusta saatava näytteistys oli kymmenen minuuttia. Data ei ollut siten suoraan käyttökelpoista METAR-dataan nähden, joten sitä jouduttiin muokkaamaan METAR-informaatioon paremmin soveltuvaksi. Apuna tässä käytettiin *prepareSolar.sh* -skriptiä (Liite 2). Laskuri ei myöskään kyennyt tuottamaan kymmenen vuoden datasarjaa yhtenä tiedostona. Skriptillä yhdistettiin aluksi kymmenen vuoden aikasarja kolmesta erillisestä tiedostosta, jonka jälkeen siitä



poistettiin kaikki muut paitsi :20- ja :50 -päätteiset aikaleimat. Näin aikaleimat saatiin täsmäämään METAR-tiedotteisiin.

Kuva 6 näyttää skriptiin sisäänkirjoitetun laskurin avulla käytettyjen työkalujen tehokkuuden riveittäin käsiteltäviin suuriin tekstidatamassoihin. Alkuperäiset 578595 riviä supistuu muutamissa sekunneissa 192864 riviin.

```
Combining solar data...
Solar data combined in 0 seconds.

Validating solar data...
Solar data validated in 1 seconds.
Data validation finished in 1 seconds.

Processing solar date and time format...
Solar datas validated in 1 seconds.

Solar Data processed and results written in 2 seconds.

user@users-Mac-mini Kandi %
```

Kuva 6. prepareSolar.sh BASH-skriptin komentorivin tulostus.

Kuva 7 vastaavasti näyttää toteen oikeellisten aikaleimojen aurinkodatan sekä METAR-datan eri näytemäärän. Oikeellisesta aurinkodatasta voitiin todeta, että aikaleimoja tulisi tarkastelujaksolla olla kaikkiaan 192864, kun taas METAR-datassa aikaleimoja (monikertojen poiston jälkeen) oli 191200. Ts., METAR-datassa oli 1664 viestin vaje. Jotta datasarjat voitaisiin yhdistää, täytyisi niiden olla samanmittaiset ja aikaleimojen täsmätä.

```
user@users-Mac-mini Kandi % awk 'END{print NR}' ./data/solar_valid.txt
192864
user@users-Mac-mini Kandi % awk 'END{print NR}' ./data/EF0U_valid.txt
191200
user@users-Mac-mini Kandi %
```

Kuva 7. METAR-datan ja aurinkodatan rivimäärien vertailua.

#### 4.4. METAR-datan ja aurinkodatan yhdistäminen

METAR-datan puuttuvat 1664 aikaleimaa tuli korjata. Aluksi tuli selvittää, miten korjaus voitaisiin teknisesti toteuttaa ja toiseksi, täytyi valita minkä taktiikan mukaisesti aukkokohdat korjattaisiin.

AWK-kieltä yksinkertaisempi vaihtoehto oli käyttää *cut*- ja *comm*-käskyjä, joiden avulla aluksi *cut*-käskyllä erotettiin METAR- ja aurinkodatan aikaleimat muusta datasta. Sen jälkeen näitä METAR- ja aurinkodatan aikaleimoja verrattiin *comm*-käskyllä. Näin aikaan saatiin *gapsMETAR.txt* -tiedosto, josta löytyivät aikaleimat kaikille METAR-datasta puuttuvalle 1664 riville. Kuva 8 esittää listauksen METAR-datasta puuttuvien aikaleimojen alkupäästä (20 näytettä vuodelta 2010) sekä loppupäästä (kymmenen viimeistä vuodelta 2020). Lisäksi on tulostettu rivinumerot kolmanteen sarakkeeseen AWK-kielen sisäänrakennettua NR-laskuria hyödyksi käyttäen. Rivinnumero on kuvassa merkitty punaisella värillä.

```

user@users-Mac-mini Kandi%awk '$3==" "{print $0, NR}' ./data/EFOU_valid.txt
2010-01-04      10:20    165
2010-01-08      19:50    376
2010-01-13      00:20    577
2010-01-13      05:20    587
2010-01-14      03:50    632
2010-01-14      23:20    671
2010-01-14      23:50    672
2010-01-15      20:20    713
2010-01-20      10:50    934
2010-01-21      06:50    974
2010-01-26      21:20   1243
2010-01-28      10:50   1318
2010-01-31      12:20   1465
2010-02-12      08:50   2034
2010-02-22      09:50   2516
2010-02-23      12:20   2569
2010-02-28      22:20   2829
2010-02-28      22:50   2830
2010-02-28      23:20   2831
2010-02-28      23:50   2832
...
2020-08-21      23:50  186528
2020-08-22      00:20  186529
2020-08-22      00:50  186530
2020-08-22      01:20  186531
2020-08-22      01:50  186532
2020-08-22      02:20  186533
2020-08-27      19:50  186808
2020-09-28      06:20  188317
2020-10-19      11:20  189335
2020-10-19      12:20  189337

```

Kuva 8. Esimerkkejä tyhjästä METAR-datan aikaleimoista ja rivinnumeroista.

Seuraava haaste oli tunnistettujen tyhjien aikaleimojen syöttäminen oikeille paikoilleen. Iowa Environmental Mesonet’in alkuperäinen aikaformaatti ei ole AWK-kielen tai UNIX-pohjaisten käskykantojen tunnistettavissa. Täten vuodet, kuukaudet, päivät, tunnit ja minuutit muokattiin AWK-kieltä hyväksikäyttäen viiteen peräkkäiseen sarakkeeseen. Näin saatiin muodostettua ikään kuin viisi rullaavaa laskuria. Muokkaus toteutettiin sekä METAR-datalle että *gapsMETAR.txt* -tiedoston sisältämille riveille. METAR-datan ja *gapsMETAR.txt* -tiedoston yhdistämiseen käytettiin *sort*-käskyä, jolla aiemmin muodostettua viittä laskurisaraketta voitiin juoksuttaa rinnakkain ja sijoittaa näin rivit oikeaan järjestykseen. METAR-datasta löytyi nyt 192864 riviä. 1664 niistä oli ilman pilvisuusdataa. Skripti *patchMETARemptyDates.sh* on esitetty kokonaisuudessaan liitteessä 3.

Seuraavaksi tuli valita taktiikka, jolla tyhjät rivit täytetään. METAR-tiedotteiden säähavainnot ovat aina keskiarvoistettuja havaintoja. Yksittäisten puuttuvien näyttöiden tapauksessa läpinäkyvin korjaustapa voisi olla täyttäminen edellisen ja seuraavan näytteen keskiarvolla. Varsin yleistä kuitenkin oli, että näytteitä puuttui useampia kuin yksi; toisaalta pilvipeite voi muuttua huomattavasti nopeammin kuin mikä on METAR-tiedotteiden lähetysväli; kolmanneksi METAR-tiedotteita käytetään oletuksella, että viimeisin on ”paras arvio tilanteesta”. Näiden pohjalta päädyttiin täyttämään tyhjät kohdat olettamalla, että viimeisin tunnettu pilvisyystilanne säilyy sellaisenaan.

Kuva 9 esittää skriptin *fillEFOUgaps.sh* komentorivitulostuksen, sekä tyhjien rivien testauksen. Skriptin ajon jälkeen tyhjiä rivejä ei enää esiintynyt. Skripti on kokonaisuudessaan esitetty liitteessä 4 ja sen avulla 1664 tyhjää aikaleimaa täytettiin viimeisellä tunnetulla METAR-tiedotteella.

```

user@users-Mac-mini Kandi % bash filleFOUgaps.sh

1664 empty METAR lines found.
Gaps fetched in 0 seconds.
Searching for yet blank data occurrences in fix data.

947 yet empty fix lines found.
Filling rest of the gaps in EFOU_fix_gapless.txt.
  <-- If void, fix data ready.
Ready in 8 seconds.
1664 gap fills created.

Rearranging date and time of EFOU data...
Ready in 1 seconds.

Rearranging fix data...
Ready in 0 seconds.

Combining EFOU data and fix data...

All METAR data processed.
Ready in 2 seconds.

user@users-Mac-mini Kandi % awk '$6==""{print $0, NR}' ./data/EFOU_valid.txt
user@users-Mac-mini Kandi %

```

Kuva 9. Tyhjien aikaleimojen täyttö sekä tarkistus mahdollisten tyhjien datarivien varalta.

Käsittelyn nopeuden testaamiseksi tyhjien datarivien täyttäminen suoritettiin kahdessa vaiheessa: ensin käytettiin AWK-kielen kahta sisäänrakennettua rivilaskuria NR ja FNR. Loput täytettiin BASH-luupissa.

```

2010-01-04      10:20      1 1.75
2010-01-08      19:50      0 0
2010-01-13      00:20      0 0
2010-01-13      05:20      0.1875 0.1875
2010-01-14      03:50      1 1
2010-01-14      23:20      1 1
2010-01-14      23:50
2010-01-15      20:20      1 1
2010-01-20      10:50      0.9375 0.9375
2010-01-21      06:50      0 0
2010-01-26      21:20      0 0
2010-01-28      10:50      0 0
2010-01-31      12:20      1 1.375
2010-02-12      08:50      0.1875 0.1875
2010-02-22      09:50      0 0
2010-02-23      12:20      0.625 0.625
2010-02-28      22:20      0.75 0.75
2010-02-28      22:50
2010-02-28      23:20
2010-02-28      23:50
2010-03-05      00:20      0 0
2010-03-06      18:50      0.1875 0.1875
2010-03-07      05:50      0 0
2010-03-08      10:50      0.1875 0.1875
2010-03-09      05:50      0 0
2010-03-14      19:50      0 0
2010-03-14      23:20      1 1
2010-03-14      23:50
2010-03-20      04:50      1 1.1875
2010-03-22      10:20      0 0

```

Kuva 10. Tiedoston EFOU\_fix.txt rivit 0-30.

Kuva 10 listaa 30 ensimmäistä riviä käsittelyn aputiedostona käytetyn *EFOU\_fix.txt* sisällöstä. Kaikki aikaleimat ovat olleet alun perin tyhjiä ja ne rivit, joista pilvisyystieto löytyy (merkitty punaisella), kuuluvat ensimmäisen vaiheen täyttöön.

NR- sekä FNR-laskureiden toiminta on sama, jos käsitellään yhtä ainoaa tiedostoa, mutta kahden tai useamman tiedoston käsittelyssä niiden toimintalogiikat eroavat. Yksinkertaistaen NR tarkoittaa ”tähän asti nähtyjen rivien lukumäärää” ja FNR samaa kyseisen tiedoston sisällä. Tyhjän rivin löytyessä, siihen kopioitiin edellisen rivin tiedot. Tämä toimii hyvin yksittäisten tyhjiä rivien tapauksessa. Jäljelle jääneet 947 tyhjää riviä – jotka siis esiintyivät useampana peräkkäisenä tyhjänä rivinä peräkkäin – korjattiin BASH-luupilla.

Kuva 11 esittää vastaavan listauksen toisen aputiedoston *EFOU\_fix\_gapless.txt* sisällöstä jälkimmäisen täyttötaktiikan jälkeen. Jäljelle jääneet ja tällä kierroksella korjatut on jälleen merkitty punaisella.

|            |       |               |
|------------|-------|---------------|
| 2010-01-04 | 10:20 | 1 1.75        |
| 2010-01-08 | 19:50 | 0 0           |
| 2010-01-13 | 00:20 | 0 0           |
| 2010-01-13 | 05:20 | 0.1875 0.1875 |
| 2010-01-14 | 03:50 | 1 1           |
| 2010-01-14 | 23:20 | 1 1           |
| 2010-01-14 | 23:50 | 1 1           |
| 2010-01-15 | 20:20 | 1 1           |
| 2010-01-20 | 10:50 | 0.9375 0.9375 |
| 2010-01-21 | 06:50 | 0 0           |
| 2010-01-26 | 21:20 | 0 0           |
| 2010-01-28 | 10:50 | 0 0           |
| 2010-01-31 | 12:20 | 1 1.375       |
| 2010-02-12 | 08:50 | 0.1875 0.1875 |
| 2010-02-22 | 09:50 | 0 0           |
| 2010-02-23 | 12:20 | 0.625 0.625   |
| 2010-02-28 | 22:20 | 0.75 0.75     |
| 2010-02-28 | 22:50 | 0.75 0.75     |
| 2010-02-28 | 23:20 | 0.75 0.75     |
| 2010-02-28 | 23:50 | 0.75 0.75     |
| 2010-03-05 | 00:20 | 0 0           |
| 2010-03-06 | 18:50 | 0.1875 0.1875 |
| 2010-03-07 | 05:50 | 0 0           |
| 2010-03-08 | 10:50 | 0.1875 0.1875 |
| 2010-03-09 | 05:50 | 0 0           |
| 2010-03-14 | 19:50 | 0 0           |
| 2010-03-14 | 23:20 | 1 1           |
| 2010-03-14 | 23:50 | 1 1           |
| 2010-03-20 | 04:50 | 1 1.1875      |
| 2010-03-22 | 10:20 | 0 0           |

Kuva 11. Tiedoston *EFOU\_fix\_gapless.txt* rivit 0-30.

Kuva 9:n listaus osoittaa ensimmäisen täyttöstrategian huomattavasti tehokkaammaksi. 717 riviä täytyivät testikoneella skriptin ja järjestelmän kutsujen aiheuttamine viiveineenkin alle sekunnissa. Vastaavasti vielä tyhjäksi jääneiden 947 rivin täyttäminen BASH-luupissa kesti pyöreästi 8 sekuntia. AWK asettaa sisäiset laskurit NR sekä FNR itsenäisesti, eikä niitä voi manipuloida. Käsittelyn nopeuttamiseksi ja skriptin yksinkertaistamiseksi tulisi käyttää apumuuttujia laskureina tai vaihtoehtoisesti hyvin useita aputiedostoja. Tämän työn kannalta riittävä käsittelynopeus saavutettiin näinkin. Samalla tuli verrattua karkealla tasolla AWK-kielen tehokkuutta BASH-luupin vastaavaan.

Kuva 12 esittää listauksen METAR-sanomien riveistä 632-672. Kuva 8 listasi puutteellisia rivejä, eli alun perin täysin puuttuvia rivejä, jotka korjattiin eheällä aikasarjalla ja joista tuli siten ”päivämäärällisiä tyhjiä rivejä”. Kyseisellä välillä punaisella merkityt rivit 632, 671 sekä 672 olivat alun perin tyhjiä. Tässä ne on täytetty

*fileFOUgaps.sh* -skriptillä viimeisimmän tunnetun pilvisyydsarvon mukaisesti. Tämän esimerkkikatkelman aikana taivas on pysynyt täysin pilvipeitteisenä. Sarakkeessa kuusi nähdään laskennallinen kokonaispeittoisuus arvoväliltä [0,1]. Viimeisessä sarakeessa nähdään laskennallisia paikoittain suurempia lukuja, sillä tässä arvoaluetta ei ole rajattu. Kuten luvussa 2.4 Taivaan peittoisuus todettiin, numeroa yksi suuremmat luvut oletetaan numeroksi yksi, sillä taivas ei voi olla enemmän kuin täysin pilvipeitteinen.

```

user@users-Mac-mini Kandi % awk 'NR>=632&&NR<=672' ./data/EF0U_valid.txt
2010 01 14 03 50 1 1
2010 01 14 04 20 1 1
2010 01 14 04 50 1 1.75
2010 01 14 05 20 1 1.75
2010 01 14 05 50 1 1
2010 01 14 06 20 1 1
2010 01 14 06 50 1 1
2010 01 14 07 20 1 1
2010 01 14 07 50 1 1
2010 01 14 08 20 1 1
2010 01 14 08 50 1 1.75
2010 01 14 09 20 1 1.75
2010 01 14 09 50 1 1.75
2010 01 14 10 20 1 1
2010 01 14 10 50 1 1
2010 01 14 11 20 1 1
2010 01 14 11 50 1 1
2010 01 14 12 20 1 1
2010 01 14 12 50 1 1
2010 01 14 13 20 1 1
2010 01 14 13 50 1 1
2010 01 14 14 20 1 1
2010 01 14 14 50 1 1
2010 01 14 15 20 1 1
2010 01 14 15 50 1 1
2010 01 14 16 20 1 1
2010 01 14 16 50 1 1
2010 01 14 17 20 1 1
2010 01 14 17 50 1 1
2010 01 14 18 20 1 1
2010 01 14 18 50 1 1
2010 01 14 19 20 1 1.75
2010 01 14 19 50 1 1
2010 01 14 20 20 1 1
2010 01 14 20 50 1 1
2010 01 14 21 20 1 1.4375
2010 01 14 21 50 1 1
2010 01 14 22 20 1 1
2010 01 14 22 50 1 1
2010 01 14 23 20 1 1
2010 01 14 23 50 1 1
user@users-Mac-mini Kandi %

```

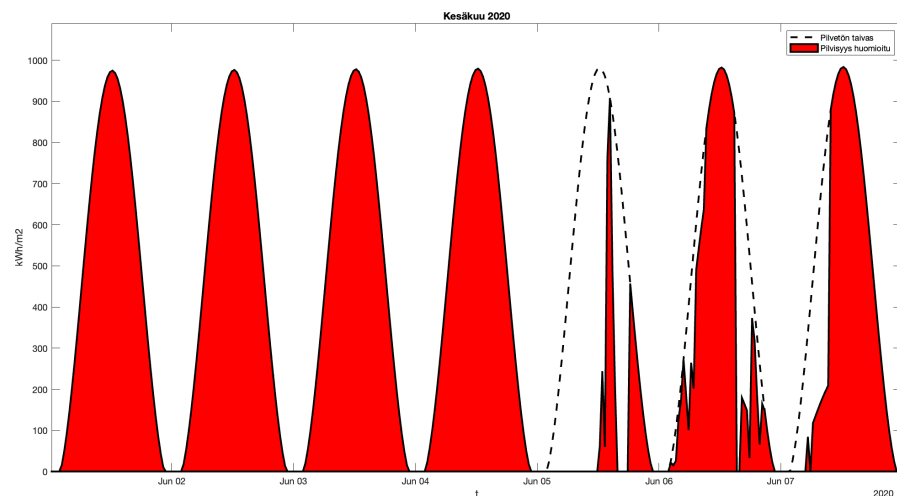
Kuva 12. Täytetyt rivit väliltä 632-672.

Tämän kandidaatintyön ydinsisältöä olivat teoriaosana METAR-datan rakenteen analysointi ja peitteisyyden laskeminen, sekä sen konstruktiivisena osana toteutukseen tarvittava koodaus. On mielekästä antaa kuitenkin esimerkki myös määritettyjen pilvisyydetietojen ja aurinkodatan yhdistämisestä komentotulkissa sekä graafinen esitys yksinkertaisesta pilvisyydetiedon sekä aurinkodatan yhdistämisestä. Kuva 13 esittää esimerkinomaisesti nopean tavan yhdistää nyt käsitelty Oulun lentoaseman pilvisyydetiedot sekä aurinkodata. Tällä tavoin jo käsiteltyyn METAR-dataan ilmestyy kahdeksas sarake, joka sisältää aurinko-osuuden.

```
user@users-Mac-mini Kandi % awk '{print $3}' ./data/solar_valid.txt > solar_data.txt
user@users-Mac-mini Kandi % paste ./data/EFOU_valid.txt ./solar_data.txt > ./data/EFOU_with_solar.txt
```

Kuva 13. Pilvisyystietojen ja aurinkodatan yhdistäminen komentotulkuissa.

Kuva 14 esittää Matlabilla työstetyn graafisen esimerkin kesäkuun 2020 ensimmäiseltä seitsemältä päivältä. Neljän ensimmäisen päivän ajan taivas on ollut selkeä (CAVOK tai SKC) ja pilvipeitteisyys siten lukuarvolla nolla. Seuraavina kolmena päivänä pilvipeitteisyys vaihtelee täydestä peitteestä (peitteisyys = 1) täysin selkeään ja näiden välillä.



Kuva 14. Pilvisyystietojen ja aurinkodatan yhdistäminen graafiseksi esitykseksi Matlabissa.

Tässä työssä käsitelty METAR-data ja siitä lasketun pilvisyystiedon jatkojohdyntämien jää mahdolliseksi jatkotyöksi.

#### 4.5. Pohdinta

Työssä käytetyt työkalut osoittautuivat erittäin tehokkaiksi välineiksi jopa satoja tuhansia rivejä sisältävien tiedostojen käsittelyyn ja parserointiin. Suuresta datamäärästä voitiin kohtuullisella vaivalla etsiä oleellinen ja karsia kaikki ylimääräinen pois. Hieman pidemmälle menevällä koodauksella onnistuivat myös useamman tiedoston korjaamiset, vertailut ja sovittamiset toisiinsa. Paikoittain tämä vaati yllättävän syvälle yksittäisen käskyn olemukseen ja toimintaan perehtymistä; paikoitellen erilaisten käskyjen skriptiksi yhdistäminen vaati jo varsin luovia ratkaisuja. Tämä johtui hyvin erilaisten yksittäisten – mutta ei välttämättä yksinään tehokkaasti toimivien – työkalujen hyvin erilaisen toimintaperiaatteen yhteensovittamisesta. Toimiminen yksittäisen kielen rajoissa on monesti selkeämpää ja suoraviivaisempaa.

Edellä mainitusta tehokkuudesta johtuen, työssä käytetyt työkalut ovat edelleen ahkerassa käytössä kaikkein tehokkainta käsittelyä vaativassa tekstitiedostojen muokkauksessa. Toinen syy lienee, että ne ovat UNIX-pohjaisten järjestelmien vakioituneita työkaluja. Ne ovat helposti saatavilla. Samalla on todettava, että tässä käytettyjen työkalujen käyttö ei välttämättä ole kovin intuitiivista, saati kovin helposti lähestyttävää. Toisaalta nimenomaan niiden tehokkuus houkuttelee niiden käyttöön ja hieman omintakeisen toimintalogiikan opettelemiseen.

#### **4.6. Jatko- ja parannuskehitysmahdollisuudet**

Käytettyjen työkalujen tehokkuus tuli työtä tehdessä todistettua. Niiden täyttä potentiaalia rajatussa kandintyössä ei kuitenkaan päästy vielä käyttämään. Kymmenen vuoden pilvisyystiedot ja lopputuloksen lähes 200000 riviä käsiteltyä dataa on valtava määrä esimerkiksi Matlabin työstettäväksi. Käytettyjä työkaluja voisi hyödyntää esimerkiksi erinäisten välitulosten tai vaikkapa vuosittaisten kertymien laskennassa; laskennassa, jossa esimerkiksi mainittu Matlab ei ole muistinhallinnallisesti tehokkaimmillaan.

Nyt suodatettua ja laskettua pilvisyystietoa olisi luonnollisesti mielenkiintoista soveltaa osana suurempaa tai kokonaisvaltaisempaa mallia. Jo nyt määritettyjen pilvisyystietojen avulla voitaisiin lähteä arvioimaan aurinkoenergiapotentiaalia Oulun lentoaseman läheisyydessä. Pienillä lisäyksillä käsillä olevasta METAR-datasta voitaisiin arvioida myös tuulitietoja. Näiden pohjalta lähialueen tuulienergiapotentiaaliakin kyettäisiin haarukoimaan.

## 5. YHTEENVETO

Työssä arvioitiin pilvipeitteisyyttä Oulun lentoaseman tuottamasta METAR-datasta käyttäen tekstitiedostojen käsittelyyn suunniteltua AWK-kieltä sekä UNIX-tyyppisen komentorivin perustyökaluja kuten *grep*, *cat*, *sort*, ja *tr*. Näistä koottiin BASH-skriptejä, joilla toiminta voitiin automatisoida (liitteet 1-4).

Perustana käytettiin Iowa Environmental Mesonet'in (IEM) kokoamaa METAR-tiedotekantaa, josta ladattiin Oulun lentoaseman kaikki METAR-tiedotteet vuosilta 2010-2020. Kokonaisuudessaan tämä käsitti yhteensä 191200 riviä tiedotteita, joiden rakenne ja pituus vaihtelivat viestistä toiseen. Tämä teki pilvisyyden määrittämisestä haasteellista. Koska METAR-tiedote on lähtökohtaisesti luotu ihmisen luettavaksi ja nopeasti sisäistettäväksi, pilvisyyden määrä oli koodattu tätä ajatellen helpompaan kirjainlyhennemuotoon, joka vastaavasti on epäkäytännöllinen matemaattisen laskennan kannalta.

Aluksi tiedotteista suodatettiin riveittäin pilvisyyden astetta kuvaavat termit ja ne koodattiin vastaaviksi numeroarvoiksi. Näiden osapeitteisyyksien avulla määritettiin taivaan kokonaispeitto. Tässä vaiheessa havaittiin, että IEM'in tietokannasta ei löydy kaikkia teoreettisesti lähetettyjä tiedotteita (kaikkiaan 1664 uupui). Todennäköisiä syitä ovat olleet laitteiden hetkittäinen vikatila tai vaikkapa huoltokatkokset. Tällaisten aukkojen kohdat paikannettiin käyttäen apuna oikeellista aikasarjaa. Tämän lisäksi edellä mainittuja työkaluja käytettiin hyväksi näin syntyneiden tyhjien tiedotteiden täyttämässä oletuksella, että viimeisin METAR-tiedote on edelleen voimassa. Näin kokonaistiedotemäärä kasvoi 192864 tiedotteeseen ja aikajanasta tuli yhtenäinen.

Edellä kuvatulla tavalla saatiin Oulun lentoaseman vuosien 2010-2020 METAR-tiedotteista muodostettua kokonaispilvisyystiedot sisältävä ehjä ja korjattu datasarja. Tällaista datasarjaa voitaisiin hyödyntää esimerkiksi paikallisen aurinkoenergiamäärän arviointiin. Lisäksi samasta METAR-datakannasta voitaisiin samoja työkaluja ja lähes samoja skriptejä käyttäen arvioida vaikkapa alueen tuulisuutta ja sitä kautta tuulienergian määrää.



## 6. LÄHTEET

- [1] Ilmatieteenlaitos, Lentosääpalvelut Suomessa (luettu: 11.02.2021), URL: [https://ilmailusaa.fi/pdf/Lentosaapalvelut\\_Suomessa\\_01-2021.pdf](https://ilmailusaa.fi/pdf/Lentosaapalvelut_Suomessa_01-2021.pdf)
- [2] ANS Finland, Ilmailukäsikirja, GEN 3.5 (luettu: 11.02.2021). URL: <https://www.ais.fi/fi/tuotteet-ja-palvelut/aip>
- [3] National Oceanic and Atmospheric Administration, FCM-H1-2017 (luettu: 15.2.2021) URL: [ftp://ftp.library.noaa.gov/noaa\\_documents.lib/OFCM/meteorological\\_handbooks/FCM\\_H1\\_2017.pdf](ftp://ftp.library.noaa.gov/noaa_documents.lib/OFCM/meteorological_handbooks/FCM_H1_2017.pdf)
- [4] Ilmatieteenlaitos, “Sääennätyksiä” (luettu: 20.9.2021) URL: <https://www.ilmatieteenlaitos.fi/tuuliennatyksia>
- [5] Vaisala CL31, ceilometrin datalehti (luettu: 27.9.2021) URL: <https://www.vaisala.com/sites/default/files/documents/CL31-Datasheet-B210415EN.pdf>
- [6] Iowa Environmental Mesonet, ASOS/METAR -data (luettu: 1.4.2021) URL: [https://mesonet.agron.iastate.edu/request/download.phtml?network=FI\\_\\_ASOS](https://mesonet.agron.iastate.edu/request/download.phtml?network=FI__ASOS)
- [7] National Renewable Energy Laboratory (NREL), solar position and intensity (SOLPOS) -laskuri (luettu: 29.9.2021) URL: <https://midcdmz.nrel.gov/solpos/solpos.html>

## 7. LIITTEET

- Liite 1 METAR-tiedotteiden valmistelu ja esikäsittely (*prepareMETAR.sh*)
- Liite 2 Aurinkodatan ja aikasarjan valmistelu (*prepareSolar.sh*)
- Liite 3 METAR-aikasarjan korjaus (*patchMETAREmptyDates.sh*)
- Liite 4 Tyhjien METAR-rivien täyttö (Liite 4, *fillEFOUgaps.sh*)

```

#!/bin/bash
#
clear
echo ""
echo "Validating data..."
SECONDS=0
awk 'BEGIN {FS=" ";OFS="t"/:20;50/" /RawData/EFOU.txt > ./data/EFOU_valid.txt
echo "Data validated in $SECONDS seconds."
let "TOTAL = $SECONDS"

echo ""
echo "Processing data..."
SECONDS=0

awk 'BEGIN {FS=OFS="t"} /VV/{ print "1", $0 };/VV/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS="t"} /OVC/{ print "1", $0 };/OVC/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS="t"} /BKN/{ print "0.75", $0 };/BKN/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS="t"} /SCT/{ print "0.4375", $0 };/SCT/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS="t"} /FEW/{ print "0.1875", $0 };/FEW/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data
./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS="t"} /SKC/{ print "0", $0 };/SKC/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS="t"} /CAVOC/{ print "0", $0 };/CAVOC/{ print "0", $0 }' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data
./data/EFOU_valid.txt
awk 'BEGIN {FS=OFS=" " } {gsub(/AUTO/,"");print}' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk 'BEGIN {FS=" ";OFS="t"} {gsub(/EFOU/,"");print}' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
tr -s 't' < ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
awk -f ./scripts/sky_coverage.awk ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt

echo "Sky coverage processed in $SECONDS seconds."

echo ""
echo "Processing METAR data for duplicates and multiple entries..."
SECONDS=0
awk '{print $1}' ./data/EFOU_valid.txt > ./data/EFOU_dates.txt
awk 'BEGIN {FS=OFS="."} {printf "%s\t%s\t%s\n", $1, $2, $3}' ./data/EFOU_dates.txt > ./temp.data && mv ./temp.data ./data/EFOU_dates.txt
awk '{print $2}' ./data/EFOU_valid.txt > ./data/EFOU_times.txt
awk 'BEGIN {FS=OFS="."} {printf "%s\t%s\n", $1, $2}' ./data/EFOU_times.txt > ./temp.data && mv ./temp.data ./data/EFOU_times.txt
awk '{print $3, $4}' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
paste ./data/EFOU_dates.txt ./data/EFOU_times.txt ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
rm ./data/EFOU_dates.txt
rm ./data/EFOU_times.txt

awk 'a[$1$2$3$4$5]++' ./data/EFOU_valid.txt > ./data/EFOU_multiple_entries.txt
[ -s ./data/EFOU_multiple_entries.txt ]
awk 'END {print NR}' ./data/EFOU_multiple_entries.txt > ./empty
empty=${<empty}

echo "Empty multiple lines found..."

while [ $empty -gt 0 ]
do
awk 'BEGIN {FS=OFS="t"} {print $1, $2, $3, $4, $5}' ./data/EFOU_multiple_entries.txt > ./data/EFOU_multiple_timedata.txt
grep -v -f ./data/EFOU_multiple_timedata.txt ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
# Use timedata again to point saved last entries and fill them by $1\t$2\t$3=date and \t$4\t$5=time to original data.
grep -f ./data/EFOU_multiple_timedata.txt ./data/EFOU_multiple_entries.txt | cat - ./data/EFOU_valid.txt | sort -n -k1 -k2 -k3 -k4 -k5 > ./temp.data && mv
./temp.data ./data/EFOU_valid.txt

awk 'a[$1$2$3$4$5]++' ./data/EFOU_valid.txt > ./data/EFOU_multiple_entries.txt
awk 'END {print NR}' ./data/EFOU_multiple_entries.txt > ./empty
empty=${<empty}
echo "Empty unprocessed multiple lines remains..."
done

rm ./empty
rm ./data/EFOU_multiple_entries.txt
rm ./data/EFOU_multiple_timedata.txt

echo ""
echo "Fixing UTC-0 timezone to UTC+2."
awk '{if ($1 != "2021") {print $1, $2, $3, $4, $5} else {print ""}}' ./data/EFOU_valid.txt > ./data/EFOU_dates_all_temp.txt
awk 'NR>4 {print $6, $7}' ./data/EFOU_valid.txt > ./data/EFOU_skipped_four_first_lines.txt
paste -d't' ./data/EFOU_dates_all_temp.txt ./data/EFOU_skipped_four_first_lines.txt > ./data/combine.txt
awk 'BEGIN {OFS="t"} {if ($1 > "2009") {print $1, $2, $3, $4, $5, $6, $7}}' ./data/combine.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
rm ./data/EFOU_dates_all_temp.txt
rm ./data/EFOU_skipped_four_first_lines.txt
rm ./data/combine.txt

echo "Timezone fixed."
echo "Recreating original data and time format."

awk 'BEGIN {FS=OFS="t"} {print $1, $2, $3}' ./data/EFOU_valid.txt > ./data/EFOU_recreate_dates.txt
awk 'BEGIN {FS="t"} {printf "%s-%s-%s\n", $1, $2, $3}' ./data/EFOU_recreate_dates.txt > ./temp.data && mv ./temp.data ./data/EFOU_recreate_dates.txt
awk 'BEGIN {FS=OFS="t"} {print $4, $5}' ./data/EFOU_valid.txt > ./data/EFOU_recreate_times.txt
awk 'BEGIN {FS="t"} {printf "%s:%s\n", $1, $2}' ./data/EFOU_recreate_times.txt > ./temp.data && mv ./temp.data ./data/EFOU_recreate_times.txt
awk '{print $6, $7}' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
paste ./data/EFOU_recreate_dates.txt ./data/EFOU_recreate_times.txt ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt
rm ./data/EFOU_recreate_dates.txt
rm ./data/EFOU_recreate_times.txt

echo ""
echo "Data processed and results written in $TOTAL seconds."
echo ""

```

```

#!/bin/bash
#

clear

echo ""
echo "Combining solar data..."
SECONDS=0
cat ./RawData/solar_2010-2013.txt ./RawData/solar_2014-2019.txt ./RawData/solar_2020.txt > ./RawData/solar_2010-2020.txt
echo "Solar data combined in $SECONDS seconds."
let "TOTAL = $TOTAL+$SECONDS"

echo ""
echo "Validating solar data..."
SECONDS=0
awk 'BEGIN {FS=" ";OFS=" "};/20;/:50/' ./RawData/solar_2010-2020.txt > ./data/solar_valid.txt
awk 'BEGIN {FS=" ";OFS=" "};{print($1, $2, $4, $3, $5)}' ./data/solar_valid.txt > ./data/temp.data && mv ./data/temp.data ./data/solar_valid.txt
#cp solar_valid.txt ./results/
echo "Solar data validated in $SECONDS seconds."
let "TOTAL = $SECONDS"
echo "Data validation finished in $SECONDS seconds."
echo ""
echo "Processing solar date and time format..."
SECONDS=0
#cp ./results/solar_valid.txt ./solar_valid.txt

awk 'BEGIN {FS=OFS="\t"}{print($1)}' ./data/solar_valid.txt > ./temp.data && mv ./temp.data ./data/solar_dates.txt
awk 'BEGIN {FS=" ";OFS=" ";}{printf "%04s-%02s-%02s\n", $3, $1, $2}' ./data/solar_dates.txt > ./temp.data && mv ./temp.data ./data/solar_dates.txt
#awk 'BEGIN {FS=" ";OFS=" "};{printf "%s\t%02d\t%02d\n", $3, $1, $2}' ./data/solar_dates.txt > ./temp.data && mv ./temp.data ./data/solar_dates.txt
awk 'BEGIN {FS="\t"}{print($2)}' ./data/solar_valid.txt > ./temp.data && mv ./temp.data ./data/solar_times.txt
awk 'BEGIN {FS=" ";}{printf "%02s:%02s\n", $1, $2}' ./data/solar_times.txt > ./temp.data && mv ./temp.data ./data/solar_times.txt
#awk 'BEGIN {FS=" ";OFS=" "};{printf "%02d\t%02d\n", $1, $2}' ./data/solar_times.txt > ./temp.data && mv ./temp.data ./data/solar_times.txt
awk 'BEGIN {FS=OFS="\t"}{print($3)}' ./data/solar_valid.txt > ./temp.data && mv ./temp.data ./data/solar_data.txt

paste ./data/solar_dates.txt ./data/solar_times.txt ./data/solar_data.txt > ./data/solar_valid.txt
rm ./data/solar_dates.txt
rm ./data/solar_times.txt
rm ./data/solar_data.txt

#awk 'BEGIN {OFS=" "};{print($1, $2, $3, $4, $5, $6)}' ./data/solar_valid.txt > ./temp.data && mv ./temp.data ./data/solar_valid.txt

echo "Solar datas validated in $SECONDS seconds."

let "TOTAL = $TOTAL + $SECONDS"
echo ""
echo "Solar Data processed and results written in $TOTAL seconds."
echo ""

```

```

#!/bin/bash
#
#Script for fitting METAR and solar data.
#Includes interpolating and parsing missing data.

clear

echo ""
echo "Comparing METAR and solar data time vectors..."
SECONDS=0

cut -f1-2 ./data/EFOU_valid.txt > METAR_dates_times.txt
cut -f1-2 ./data/solar_valid.txt > solar_valid_dates_times.txt
comm -3 METAR_dates_times.txt solar_valid_dates_times.txt > gapsMETAR.txt
rm METAR_dates_times.txt
rm solar_valid_dates_times.txt

# sort -attribute does not understand YYYY-MM-DD nor HH:MM time format.
# Divide YYYY-MM-DD -> YYYYtMMtDD format to enable "sort"-usage later.
awk '{print $1}' ./data/EFOU_valid.txt > EFOU_valid_dates.txt
awk 'BEGIN {FS=OFS="."} {printf "%s\t%s\t%s\n", $1, $2, $3}' EFOU_valid_dates.txt > ./temp.data && mv ./temp.data ./EFOU_valid_dates.txt
awk '{print $2}' ./data/EFOU_valid.txt > EFOU_times.txt
awk 'BEGIN {FS=OFS="."} {printf "%s\t%s\n", $1, $2}' EFOU_times.txt > ./temp.data && mv ./temp.data ./EFOU_valid_times.txt
rm EFOU_times.txt
# Remove $1 and $2 from original data.
awk '{print $3, $4}' ./data/EFOU_valid.txt > ./temp.data && mv ./temp.data ./EFOU_valid_temp.txt
# Combine new format of dates and times to other data.
paste ./EFOU_valid_dates.txt ./EFOU_valid_times.txt ./EFOU_valid_temp.txt > ./temp.data && mv ./temp.data ./EFOU_valid_temp.txt
# Remove temporary date and time files.
rm ./EFOU_valid_dates.txt
rm ./EFOU_valid_times.txt

# sort -attribute does not understand YYYY-MM-DD nor HH:MM time format.
# Divide YYYY-MM-DD -> YYYYtMMtDD format to enable "sort"-usage later.
awk '{print $1}' gapsMETAR.txt > gapsMETAR_dates.txt
awk 'BEGIN {FS=OFS="."} {printf "%s\t%s\t%s\n", $1, $2, $3}' gapsMETAR_dates.txt > ./temp.data && mv ./temp.data ./gapsMETAR_dates.txt
awk '{print $2}' gapsMETAR.txt > gapsMETAR_times.txt
awk 'BEGIN {FS=OFS="."} {printf "%s\t%s\n", $1, $2}' gapsMETAR_times.txt > ./temp.data && mv ./temp.data ./gapsMETAR_times.txt
# Remove $1 and $2 from original data.
awk '{print $3}' gapsMETAR.txt > ./temp.data && mv ./temp.data ./gapsMETAR.txt
# Combine new format of dates and times to other data.
paste ./gapsMETAR_dates.txt ./gapsMETAR_times.txt ./gapsMETAR.txt > ./temp.data && mv ./temp.data ./gapsMETAR.txt
# Remove temporary date and time files.

cat gapsMETAR.txt | cat - EFOU_valid_temp.txt | sort -n -k1 -k2 -k3 -k4 -k5 > ./temp.data && mv ./temp.data ./EFOU_valid_temp.txt
rm gapsMETAR.txt

# Recreate the original date and time formats.
# YYYYtMMtDD --> YYYY-MM-DD && HHtMM --> HH:MM
awk 'BEGIN {FS=OFS="t"} {print $1, $2, $3}' EFOU_valid_temp.txt > EFOU_valid_recreate_dates.txt
awk 'BEGIN {FS="t"} {printf "%s-%s-%s\n", $1, $2, $3}' EFOU_valid_recreate_dates.txt > ./temp.data && mv ./temp.data ./EFOU_valid_recreate_dates.txt
awk 'BEGIN {FS=OFS="t"} {print $4, $5}' EFOU_valid_temp.txt > EFOU_valid_recreate_times.txt
awk 'BEGIN {FS="t"} {printf "%s:%s\n", $1, $2}' EFOU_valid_recreate_times.txt > ./temp.data && mv ./temp.data ./EFOU_valid_recreate_times.txt
awk '{print $6, $7}' EFOU_valid_temp.txt > ./temp.data && mv ./temp.data ./EFOU_valid_temp.txt
paste ./EFOU_valid_recreate_dates.txt ./EFOU_valid_recreate_times.txt ./EFOU_valid_temp.txt > ./temp.data && mv ./temp.data ./data/EFOU_valid.txt

echo "METAR time gaps detected and empty data rows per date and time added in $SECONDS seconds."
let "TOTAL = $SECONDS"
echo ""
echo ""

```

```

#!/bin/bash
#

SECONDS=0
# If there are rows with no data, print line numbers to gapRowNR.txt.
awk 'S3=="" {print NR}' ./data/EFOU_valid.txt > gapRowNR.txt
# How many rows with gap.
cat gapRowNR.txt | wc -l | tr -d " " > howManyGapRows.txt
values=$(<gapRowNR.txt)
length=$(<howManyGapRows.txt)

echo ""
echo "Length empty METAR lines found."
# Create fix data by assuming previous data.
#####
awk 'FNR==NR {a[$1]:next}(FNR in a){print $0}' gapRowNR.txt ./data/EFOU_valid.txt > EFOU_empty.txt
awk 'FNR==NR {a[$1]:next}(FNR+1 in a){print $3, $4}' gapRowNR.txt ./data/EFOU_valid.txt > EFOU_fix.txt
paste EFOU_empty.txt EFOU_fix.txt > ./temp.data && mv ./temp.data ./EFOU_fix.txt

TOTAL=$((TOTAL+SSECONDS))
echo "Gaps fetched in $SECONDS seconds."
SECONDS=0

IFS=$'\t' # 'for item in $variable' does not work if set before.
OFS=$'\t' # 'for item in $variable' does not work if set before.

length=$(<howManyGapRows.txt)

echo "Searching for yet blank data occurrences in fix data."
# If there are rows with no data in EFOU_fix.txt, print line numbers to gapFixRowNR.txt.
awk 'S3=="" {print NR}' EFOU_fix.txt > gapFixRowNR.txt
# How many rows with gap.
cat gapFixRowNR.txt | wc -l | tr -d " " > howManyGapFixRows.txt
valuesFix=$(<gapFixRowNR.txt)
lengthFix=$(<howManyGapFixRows.txt)

echo ""
echo "LengthFix yet empty fix lines found."
echo "Filling rest of the gaps in EFOU_fix_gapless.txt."

fixDataRow=1
fixDataPointer=0

loop=$(for ((i=1; i<=length; i++)); do
    fixDataPointer=$(awk 'NR==${fixDataRow}' {print $1}' gapFixRowNR.txt

    if [[ "$i" = "${fixDataPointer}" ]]; then
        awk 'NR==${i}' {print $0}' EFOU_fix.txt
        lastValidData=$(awk 'NR==${i}' {print $3, $4}' EFOU_fix.txt)
    elif [[ "$i" = "${fixDataPointer}" ]]; then
        date=$(awk 'NR==${i}' {print $1, $2}' EFOU_fix.txt)
        printf "%s\t%s\n" "$date" "$lastValidData"
        let "fixDataRow++"
    else
        echo ""
        printf "%s" "No more empty rows found.\n"
        echo "n"
        break
    fi
done)

printf "%s\n" "$loop" > EFOU_fix_gapless.txt

# One more test if gaps exist.
anyMoreGaps=$(awk 'S3=="" {print NR}' EFOU_fix_gapless.txt)
TOTAL=$((TOTAL+SSECONDS))
echo "SanyMoreGaps <- If void, fix data ready."
fixedGaps=$(awk 'END {print NR}' EFOU_fix_gapless.txt)
echo "Ready in $SECONDS seconds."
echo "SfixedGaps gap fills created."
SECONDS=0
# Modify EFOU_valid.txt and EFOU_fix_gapless.txt date and time to a information
# that can be better used to arrange dates and time in correct order.
echo ""
echo "Rearranging date and time of EFOU data..."

awk '{print $1}' ./data/EFOU_valid.txt > EFOU_dateMod.txt
awk '{print $2}' ./data/EFOU_valid.txt > EFOU_timeMod.txt
awk '{print $3, $4}' ./data/EFOU_valid.txt > EFOU_rest.txt

awk 'BEGIN { FS = "\t"
    OFS = "\t" # make tab the field separator
    }
    {
        printf "%s\t%s\t%s\t\n", $1, $2, $3
    }' EFOU_dateMod.txt > ./temp.data && mv ./temp.data ./EFOU_dateMod.txt

temp.data ./EFOU_fix_timeMod.txt

# Listaus jatkuu seuraavalla sivulla (sivu 31)

```

```

#Liite 4: jatkoa sivun 30 listauksesta
awk 'BEGIN{ FS =":"
      OFS ="\t" # make tab the field separator
      }
      {
      printf "%s\t%s\n", $1, $2
      }' EFOU_timeMod.txt > ./temp.data && mv ./temp.data ./EFOU_timeMod.txt

paste ./EFOU_dateMod.txt ./EFOU_timeMod.txt ./EFOU_rest.txt > EFOU_data.txt
tr -s '\t' < ./EFOU_data.txt > ./data/temp.data && mv ./data/temp.data ./EFOU_data.txt
rm EFOU_rest.txt

#Field separator set as space to enable 'grep' to search with several fields. Tab separator would not work.
awk 'BEGIN{OFS=" "} { $1=$1; print $0}' EFOU_data.txt > ./data/temp.data && mv ./data/temp.data ./EFOU_data.txt

TOTAL=$TOTAL+$SECONDS
echo "Ready in $SECONDS seconds."
SECONDS=0
#####
awk '$6=""' {print $0}' EFOU_data.txt > EFOU_gaps.txt
awk '$6!=""' {print $0}' EFOU_data.txt > nonBlankEFOU.txt
#####
# Do the same for fix data.
echo ""
echo "Rearranging fix data..."

awk '{print $1}' EFOU_fix_gapless.txt > EFOU_fix_dateMod.txt
awk '{print $2}' EFOU_fix_gapless.txt > EFOU_fix_timeMod.txt
awk '{print $3, $4}' EFOU_fix_gapless.txt > EFOU_fix_rest.txt

awk 'BEGIN{ FS =":"
      OFS ="\t" # make tab the field separator
      }
      {
      printf "%s\t%s\t%s\n", $1, $2, $3
      }' EFOU_fix_dateMod.txt > ./temp.data && mv ./temp.data ./EFOU_fix_dateMod.txt

awk 'BEGIN{ FS =":"
      OFS ="\t" # make tab the field separator
      }
      {
      printf "%s\t%s\n", $1, $2
      }' EFOU_fix_timeMod.txt > ./temp.data && mv ./temp.data ./EFOU_fix_timeMod.txt

paste ./EFOU_fix_dateMod.txt ./EFOU_fix_timeMod.txt ./EFOU_fix_rest.txt > EFOU_fix_data.txt
tr -s '\t' < ./EFOU_fix_data.txt > ./data/temp.data && mv ./data/temp.data ./EFOU_fix_data.txt

#Field separator set as space to enable 'grep' to search with several fields. Tab separator would not work.
awk 'BEGIN{OFS=" "} { $1=$1; print $0}' EFOU_fix_data.txt > ./data/temp.data && mv ./data/temp.data ./EFOU_fix_data.txt

TOTAL=$TOTAL+$SECONDS
echo "Ready in $SECONDS seconds."
SECONDS=0

echo ""
echo "Combining EFOU data and fix data..."

grep -f EFOU_gaps.txt EFOU_fix_data.txt | cat - nonBlankEFOU.txt | sort -n -k1 -k2 -k3 -k4 -k5 > EFOU_final.txt

# Recreate date and time formats.
awk '{print $1, $2, $3}' EFOU_data.txt > EFOU_dateMod.txt
awk '{print $4, $5}' EFOU_data.txt > EFOU_timeMod.txt

awk '{print $6, $7}' EFOU_final.txt > EFOU_data2.txt

awk 'BEGIN{ FS =""
      OFS ="\t" # make tab the field separator
      }
      {
      printf "%s-%s-%s\n", $1, $2, $3
      }' EFOU_dateMod.txt > ./temp.data && mv ./temp.data ./EFOU_dateMod.txt

awk 'BEGIN{ FS =""
      OFS =":" # make tab the field separator
      }
      {
      printf "%s:%s\n", $1, $2
      }' EFOU_timeMod.txt > ./temp.data && mv ./temp.data ./EFOU_timeMod.txt

paste ./EFOU_dateMod.txt ./EFOU_timeMod.txt ./EFOU_data2.txt > EFOU_fixed.txt

awk 'BEGIN{OFS="\t"} { $1=$1; print $0}' EFOU_final.txt > ./data/temp.data && mv ./data/temp.data ./EFOU_fixed.txt

cp ./EFOU_fixed.txt ./data/EFOU_valid.txt

rm ./*.txt

echo ""
echo "All METAR data processed."
TOTAL=$TOTAL+$SECONDS
echo "Ready in $SECONDS seconds."
echo ""
echo ""

```